

Pixel-based data modifying method and device

FIELD OF THE INVENTION

The present invention relates to a method of modifying data in an encoded data signal, comprising at least:

- a decoding step for decoding said encoded data signal and providing a decoded data signal,
- a re-encoding step performed on a modified data signal and generating a coding error,
- a prediction step for providing a motion-compensated signal from said coding error and comprising at least a subtracting sub-step between an input data signal obtained at least from said decoded data signal and said motion-compensated signal for obtaining said modified data signal.

The invention also relates to video processing devices for carrying out said method. This invention, may be used, for example, when a broadcaster wants to introduce additional data into a sequence of coded pictures. This invention finds applications not just in the field of MPEG-2 compression, but more generally in any digital video data compression system.

BACKGROUND OF THE INVENTION

Modifying data in an encoded data signal has become a vital function in studio editing environments. A possible solution has been proposed in the international patent application WO 99/51033 (PHF98546). This patent application describes a method and its corresponding device for modifying data in an encoded data signal. This method allows an additional data signal insertion, e.g. a logo inserting, into an MPEG-2 bitstream thanks to bit rate transcoding. Logo insertion comes as an extension of the bit rate transcoder. The corresponding diagram, depicted in Fig.1, comprises a transcoding module 101 and a logo addition branch 102. The general outline of the transcoding module 101, well known to a person skilled in the art, comprises :

- a residue decoding branch 118 for receiving the input signal 125 and providing a decoded data signal Error_I(n). This branch comprises in series a variable length

decoding 107, an inverse quantization 108 followed by an inverse discrete cosine transform 109.

- a re-encoding/decoding branch 120 for providing the output signal 126 and its decoded version respectively. The re-encoding part, for providing said output signal, comprises in series a discrete cosine transform 110, a quantization 111, a variable length coding 112 followed by a buffer 113, and regulation means 114 ensuring a constant picture quality of the output signal 126, and a first subtracter 122 generating a coding error. The decoding part comprises in series an inverse quantization 115 followed by an inverse discrete cosine transform 116.
- an intermediate branch 119 comprising a motion compensation 105 using motion vectors $V(n)$ of the input signal, its associated memory 106 storing a previous signal, and a second subtracter 123. This branch, also called prediction loop, avoids the quality drift in the output signal by applying a motion compensation to said coding error generated during the re-encoding step.

The logo addition branch 102 is implemented thanks to a residue addition to the decoded signal $Error_I(n)$, by means of the adding sub-step 121. This residue is formed by subtracting an additional data signal $Logo(n)$ referenced 127 with a motion-compensated logo prediction referenced 129, obtained by means of the motion compensation sub-step 103, which is based on reference pictures containing logo previously stored in memory 104 and which uses the same vectors $V(n)$ as the main input signal.

In the prior art diagram depicted in Fig.1, two motion compensations are performed : a first one 105, well known and provided for correcting the quality drift on P and B pictures introduced by the quantization sub-step 111, and a second one 103 on an additional data signal 127. This motion compensation 103 generates said motion compensated signal $PRED(Logo(n-1), V(n))$ referenced 129 which is subtracted from said signal 127. Said motion-compensated signal is indeed essential since it cancels undesired parts of the signal relative to signal 127, previously motion-compensated by 105, in the input signal of the re-encoding step. Moreover, as a motion compensation always requires a storage of a previous signal, two memories 104 and 106 are also needed. Then, with these two motion compensation operations and two memory blocks, this solution remains not only complex as regards the CPU burden, but also expensive as regards storage memory.

SUMMARY OF THE INVENTION

It is an object of the invention to provide a method of modifying data in an encoded data signal which requires less memory capacity and puts a lesser burden on central processing units (CPU).

The method of modifying data according to the invention is characterized in that it comprises :

- a first sub-step for adding an additional data signal to said decoded data signal, for providing said input data signal,
- a second sub-step for adding said additional data signal to said coding error, said motion-compensated signal resulting from the motion compensation of the output signal of said second adding sub-step.

A variant of the previously characterized method is also proposed. It is characterized in that it comprises a sub-step for adding an additional data signal to said modified data signal, before said re-encoding step.

The corresponding diagrams, depicted in Fig.2 and Fig.3 respectively, are based on data addition in the pixel domain of the additional data signal 127 with the decoded data signal relative to the input data signal, or with signal situated in the transcoder drift correction loop, by means of said adding sub-steps.

According to the invention, in contrast to the prior art solution, no more separate motion compensation is applied to logo data since the motion compensation relative to logo data is merged with the motion compensation relative to the drift correction of the transcoder loop. The invention thus comprises a minimum number of functional sub-steps, leading to a cost-effective solution. Indeed, only one set of motion compensations and its associated memory storage is used, which simplification is possible in that advantage is taken of combinations between different sub-steps, and by using their own characteristics such as the linearity of the motion compensation.

Another object of the invention is to propose devices for carrying out the above-mentioned methods.

To this end, the invention relates in a first implementation, to a transcoding device for adding data to an encoded data signal, comprising :

- a first means for adding an additional data signal to said decoded data signal for providing said input data signal,
- a second means for adding said additional data signal to said coding error, said motion-compensated signal resulting from the motion compensation of the output signal of said second means.

In a second implementation, the invention also relates to a transcoding device for adding data to an encoded data signal, characterized in that it comprises means for adding an additional data signal to said modified data signal, before re-encoding means.

Detailed explanations and other aspects of the invention are given below.

5

BRIEF DESCRIPTION OF THE DRAWINGS

The particular aspects of the invention will now be explained with reference to the embodiments described hereinafter and considered in connection with the accompanying drawings, in which identical parts or sub-steps are designated in the same manner :

10

Fig.1 illustrates the outline, as known in the prior art, of a transcoder with its logo insertion branch,

Fig.2 illustrates a first embodiment of the technical solution according to the present invention.

15

Fig.3 illustrates a second embodiment of the technical solution according to the present invention.

DETAILED DESCRIPTION OF THE INVENTION

20

As was stated above, the present invention aims at reducing the cost of the prior art method for modifying data in an encoded data signal. Such an invention is well adapted to the case of MPEG-2 coded video signals as the input signal, but it will be apparent to a person skilled in the art that such a method is applicable to any coded signal that has been encoded with a block-based compression method such as, for example, the one described in MPEG-4, H.261 or H.263 standards. In the following, the invention will be detailed assuming that encoded video signals comply with the MPEG-2 international video standard (Moving Pictures Experts Group, ISO/IEC 13818-2).

25

30

Fig.2 depicts the first cost-effective arrangement for data insertion into an encoded data signal, according to the present invention. This arrangement re-uses the aim of the transcoder described above, into which are inserted sub-steps aiming at modifying the input signal. Indeed, the input signal is modified by a pixel-based data signal 127 simultaneously introduced owing to two adding sub-steps :

- the sub-step 121 placed at the output of the error residue decoding, more precisely on the output signal of the inverse discrete cosine transform 109. The modification of the input signal, e.g. in the case of a logo insertion, is therefore first implemented by means of an addition between the inserted data signal 127 and the incoming data

signal Error_I(n). This addition results in a signal corresponding to the positive input of the subtracting sub-step 123.

- the sub-step 124 at the input of the memory 206 relative to the motion compensation 205.

5 The modification of the input signal, e.g. in case of a logo addition, is therefore secondly implemented by means of an addition between the inserted data signal 127 and the output signal of the subtracting sub-step 122. This addition results in a signal corresponding to the input signal of the memory 206.

10 From an algorithmic point of view, this first arrangement proposed according to the invention is equivalent to the one described in the prior art of Fig.1, as it can be recursively demonstrated hereinafter.

The following notations will be adopted for the demonstration :

- V(n) : vectors of picture number n,
- I(n) : decoded input picture number n,
- 15 • Error_I(n) : error residue of input picture number n,
- O1(n) : decoded picture number n corresponding to the output signal of Fig.1,
- MEM1(n) : picture number n stored in the frame memory 106,
- O2(n) : decoded picture number n corresponding to the output signal of Fig.2,
- MEM2(n) : picture number n stored in the frame memory 206,
- 20 • PRED(X(n), V(n+1)) : motion compensation of picture X(n) using vectors V(n+1). It corresponds to a predicted version of picture X(n+1),
- T : transform defined by $T(x) = \text{IDCT}(\text{IQ}(\text{Q}(\text{DCT}(x))))$.

Note that the decoded pictures I(n), O1(n) and O2(n) are not represented by any figures since only compressed signals are accessible.

25 The equivalence between the prior art and the diagram of Fig.2 will be demonstrated if for each n, the three following relations are valid :

- 1) $O1(n) = O2(n)$
- 2) $\text{MEM1}(n) = O1(n) - I(n) - \text{Logo}(n)$
- 3) $\text{MEM2}(n) = O2(n) - I(n)$

30

Obviously, the input signal and the inserted data signal 127 of Fig.1 and Fig.2 are supposed to be identical in this demonstration.

For the case where $n = 0$, corresponding to an Intra-coded picture, it can be written :

$$4) \quad \text{Error_I}(0) = I(0)$$

It can be deduced from Fig.1 :

$$5) \quad O1(0) = T(I(0) + Logo(0))$$

$$6) \quad \text{MEM1}(0) = O1(0) - I(0) - Logo(0)$$

It can be deduced from Fig.2 :

$$7) \quad O2(0) = T(I(0) + Logo(0))$$

$$8) \quad \text{MEM2}(0) = O2(0) - I(0)$$

It is possible to conclude from relations (5), (6), (7) and (8) that relations (1), (2) and (3) are valid for $n = 0$. Let us suppose they are still valid at the rank n , and let us demonstrate that (1), (2) and (3) are also valid at the rank $(n+1)$.

Let us now introduce the terms $A(n+1)$ and $B(n+1)$ as :

$$A(n+1) = \text{Error_I}(n+1) + Logo(n+1) - \text{PRED}(Logo(n) , V(n+1)) - \text{PRED}(\text{MEM1}(n), V(n+1))$$

$$B(n+1) = \text{Error_I}(n+1) + Logo(n+1) - \text{PRED}(\text{MEM2}(n), V(n+1))$$

Since (2) and (3) are valid at the rank n , and since the motion compensation is linear, $A(n+1)$ and $B(n+1)$ become :

$$9) \quad A(n+1) = \text{Error_I}(n+1) + \text{PRED}(I(n), V(n+1)) + Logo(n+1) - \text{PRED}(O1(n) , V(n+1))$$

$$10) \quad A(n+1) = I(n+1) + Logo(n+1) - \text{PRED}(O1(n) , V(n+1))$$

$$11) \quad B(n+1) = \text{Error_I}(n+1) + \text{PRED}(I(n), V(n+1)) + Logo(n+1) - \text{PRED}(O2(n) , V(n+1))$$

$$12) \quad B(n+1) = I(n+1) + Logo(n+1) - \text{PRED}(O2(n) , V(n+1))$$

Since (1) is valid for at the rank n , relations (10) and (12) become :

$$13) \quad A(n+1) = B(n+1) = I(n+1) + Logo(n+1) - \text{PRED}(O1(n) , V(n+1))$$

It can be deduced from Fig.1 and Fig.2 :

$$14) \quad O1(n+1) = T(A(n+1)) + \text{PRED}(O1(n), V(n+1))$$

- 15) $MEM1(n+1) = T(A(n+1)) - A(n+1)$
 16) $O2(n+1) = T(B(n+1)) + PRED(O2(n), V(n+1))$
 17) $MEM2(n+1) = T(B(n+1)) - B(n+1) + Logo(n+1)$

5 One can conclude from relations (13), (15) and (17) that :

- 18) $MEM1(n+1) = O1(n+1) - I(n+1) - Logo(n+1)$
 19) $MEM2(n+1) = O2(n+1) - I(n+1)$

10 This means that relations (1), (2) and (3) are valid for the rank $n+1$, which proves the algorithmic equivalence between the arrangement of Fig.1 of the prior art, and the first proposed arrangement depicted in Fig.2 according to the invention. This proposed arrangement thus ensures that the modified output signal has the same quality as the one of the prior art but is obtained in a more cost-effective manner. Indeed, no more separate motion compensation and its associated memory is needed for the inserted data signal 127, since said data can be directly inserted into the transcoding pseudo-prediction loop, this simplification being justified by the linearity of the motion compensation. This merging of the two motion compensations - if the insertion of the two adding sub-steps 121 and 124, at no cost for most digital signal processors, is disregarded - represents a substantial gain in terms of CPU occupation as well as memory storage.

20 Fig.3 depicts an alternative embodiment of the present invention. It is also based on a transcoder arrangement identical to the one previously described and depicted in Fig.1. Compared with a transcoder architecture such as the one depicted in Fig.1, only a few modifications are made to obtain a change in the input signal. Indeed, the input signal is modified by signal 127 introduced by means of only one adding sub-step 121 placed at the input of the re-encoding step, more precisely on the input signal of the discrete cosine transform 110. The modification of the input signal, e.g. in the case of a logo addition, is therefore implemented by means of an addition between the inserted data signal 127 and the output signal of the subtracting sub-step 123. This addition results in a signal corresponding to the input of the discrete cosine transform 110. From an algorithmic point of view, this second arrangement is also equivalent to the prior art arrangement of Fig.1.

The following notations will be used for the demonstration :

- $O3(n)$: decoded picture number n corresponding to the output signal of Fig.3,
- $MEM3(n)$: picture number n stored in the frame memory 206.

Note that the decoded pictures $O3(n)$ is not represented by any figures since only compressed signals are accessible.

The same recursive demonstration can be made in proving for each n , the three following equations :

$$(20) \quad O1(n) = O3(n)$$

$$(2) \quad MEM1(n) = O1(n) - I(n) - Logo(n) \quad (\text{as demonstrated above})$$

$$(21) \quad MEM3(n) = O3(n) - I(n)$$

10 Obviously, the input signal and the inserted data signal 127 of Fig.1 and Fig.3 are supposed to be identical in this demonstration.

For the case where $n = 0$, corresponding to an Intra-coded picture, it can be written :

$$(22) \quad Error_I(0) = I(0)$$

15 It can be deduced from Fig.3 :

$$(23) \quad O3(0) = T(I(0) + Logo(0))$$

$$(24) \quad MEM3(0) = O3(0) - I(0)$$

20 From relations (5), (6), (23) and (24), it is possible to conclude that relations (20) and (21) are valid for $n = 0$. Let us suppose they are still valid at the rank n , and let us demonstrate that (20) and (21) are also valid at the rank $(n+1)$.

Let us now introduce the terms $C(n+1)$ as :

$$25 \quad C(n+1) = Error_I(n+1) + Logo(n+1) - PRED(MEM3(n), V(n+1))$$

Since (21) are valid at the rank n , and since the motion compensation is linear, $C(n+1)$ becomes :

$$25) \quad C(n+1) = Error_I(n+1) + PRED(I(n), V(n+1)) + Logo(n+1) - PRED(O3(n), V(n+1))$$

$$30 \quad 26) \quad C(n+1) = I(n+1) + Logo(n+1) - PRED(O3(n), V(n+1))$$

Since (20) is valid for the rank n , relation (26) becomes :

$$27) \quad A(n+1) = C(n+1) = I(n+1) + Logo(n+1) - PRED(O1(n), V(n+1))$$

It can be deduced from Fig.3 :

$$28) \quad O3(n+1) = T(C(n+1)) + \text{PRED}(O3(n), V(n+1))$$

$$29) \quad \text{MEM3}(n+1) = T(C(n+1)) - C(n+1) + \text{Logo}(n+1)$$

It can be concluded from relations (15), (27) and (29), that :

$$30) \quad \text{MEM3}(n+1) = O3(n+1) - I(n+1)$$

5 So relations (20) and (21) are valid for the rank $n+1$. This proposed scheme thus ensures that identical results will be obtained in the output signals of Fig.1 and Fig.3. No more separate motion compensation and its associated memory on signal 127 are needed, and said pixel-based data signal 127 is introduced thanks to the only no-cost adding sub-step ADD. In terms of CPU occupation and memory storage, this solution is almost the same as
10 the one of an isolated transcoder without data insertion, which is remarkable.

In Fig.2 and Fig.3 described above according to the invention, the input signal data is modified thanks to the insertion of the pixel-based data signal 127 by means of adding sub-steps. These inserted data may correspond to a logo, i.e. a single small picture, or a ticker, i.e. successive small different pictures. In both cases, each picture must be pixel-
15 based, e.g. by being encoded according to the so-called bitmap format which corresponds to a rough digital image coding. Of course, before insertion by means of the adding sub-steps, said signal 127 may derive from an adapted pixel-based signal Logo_ori(n) referenced 328 in order to optimize the quality of the output signal, as it is only represented in Fig.3 with step 317, for example by changing the luminance or the chrominance levels, as far as their format
20 is still compatible. It is obvious that such an adaptation does not restrict the scope and the degree of protection of the present invention.

This method of modifying data in an encoded data signal can be implemented in several manners, such as by means of wired electronic circuits or, alternatively, by means of a set of instructions stored in a computer-readable medium, said instructions replacing at
25 least a portion of said circuits and being executable under the control of a computer or a digital processor in order to carry out the same functions as fulfilled in said replaced portions. The invention then also relates to a computer-readable medium comprising a software module that includes computer executable instructions for performing the steps, or some steps, of the method described above.